Université Ibn Tofail

ENSA KENITRA

DEV C++

Programmation en C

Exercices corrigés



2ème année cycle préparatoire (S4)

Intro

Avant-propos

Le langage C a été mis au point par **D.Ritchie** et **B.W.Kernighan** au début des années 70. Leur but était de permettre de développer un langage qui permettrait d'obtenir un système d'exploitation de type UNIX portable. D.Ritchie et B.W.Kernighan se sont inspirés des langages B et BCPL, pour créer un nouveau langage : le langage C.

La première définition de ce langage a été donnée dans leur livre commun « The C programming language ». Toutefois, suite à l'apparition de nombreux compilateurs C, l'ANSI (abréviation de American National Standards Institute) a décidé de normaliser ce langage pour donner ce que l'on appelle le **C-ANSI**. Suite à cette norme, Ritchie et Kernighan ont sorti une deuxième édition du livre en intégrant les modifications apportées par l'ANSI.

Le langage C reste un des langages les plus utilisés actuellement. Cela est dû au fait que le langage C est un langage qui comporte des instructions et des structures de haut niveau (contrairement à l'assembleur par exemple) tout en générant un code très rapide grâce à un compilateur très performant.

Un des principaux intérêts du C est que c'est un langage très **portable**. Un programme écrit en C en respectant la norme ANSI est portable sans modifications sur n'importe quel système d'exploitation disposant d'un compilateur C : Windows, UNIX, VMS (système des VAX) ou encore OS/390 ou z/Os (l'OS des mainframes IBM).

La rapidité des programmes écrits en C est en grande partie due au fait que le compilateur*présuppose* que le programmeur sait ce qu'il fait : il génère un code ne contenant pas de vérifications sur la validité des pointeurs, l'espace d'adressage, etc. Ainsi, les programmes en C sont très compacts.

De plus, une des caractéristiques du C est qu'il est un langage « faiblement typé » : les types de données qu'il manipule sont très restreints, et proches de la représentation interne par le processeur : par exemple, le type 'Chaîne de caractères' n'existe pas en C. A l'inverse, comparer un entier et un caractère a un sens en C car un caractère est bien représenté en interne par le processeur par une valeur de type entier (le code ASCII ou le code EBCDIC).

Enfin et pour conclure, il est inexact que le C est un langage difficile à apprendre ! Au contraire : le C dispose de peu d'instructions, les structures de données sont limitées, etc.

Le C est un langage concis et son apprentissage est beaucoup moins ardu que ne peut l'être celui du Pascal par exemple. Voire l'exemple de code sur le JDN : Comment afficher les nombres allant de 1 à 1000 en C ou C++ sans utiliser de boucle ni de structure conditionnelle ?)

L'apprentissage du C est ainsi nécessaire pour quiconque s'intéresse à la programmation, et cet apprentissage en vaut la peine !

Table des matières

Chapitre 1 : Généralités sur le langage C

ÉNONCÉS D'EXERCICES :

Exercice 1 :

Pour convertir des degrés Fahrenheit en degrés Celsius, on a la formule suivante: C≈0.55556 × (F − 32). où F est une température en degrés Fahrenheit et C la température correspondante en degrés Celsius.

1. Écrire un programme C qui convertit une température entrée au clavier exprimée en degrés Fahrenheit et affiche une valeur approchée de la même température en degrés Celsius. Les températures seront exprimées par des nombres réels.

2. Même question qu'au 1. pour la conversion inverse : de degrés Celsius en degrés Fahrenheit.

Exercice 2 :

Lors d'une opération de promotion, un magasin de composants *hardware* applique une réduction de 10% sur tous les composants. Écrire un programme qui lit le prix d'un composant au clavier et affiche le prix calculé en tenant compte de la réduction.

Exercice 3 :

Soit la fonction mathématique f définie par f (x) = (2x + 3)*(3x^2 + 2).

1. Écrire un programme C qui calcule l'image par f d'un nombre saisi au clavier.

2. Une approximation de la dérivée f' de la fonction f est donnée en chaque point x, pour h assez petit (proche de 0), par : $f^{'}(x) = \frac{f(x+h)-f(x)}{h}$.

Écrire un programme C qui calcule et affiche une approximation de la dérivée de f en un point x entré au clavier. On pourra faire saisir le paramètre h au clavier.

Exercice 4 :

Une bille de plomb est lâchée du haut d'un immeuble et tombe en chute libre. Au bout d'un temps t (exprimé en secondes), la bille est descendue d'une hauteur (en mètres) : $\mathrm{h} = \frac{1}{2}\mathrm{gt}^2$ où $\mathrm{g} = 9.81(\mathrm{m.s}^{-2})$.

1. Écrire un programme qui calcule la hauteur descendue au bout d'un temps t saisi au clavier.

2. Écrire un programme qui calcule la durée totale de la chute connaissant la hauteur totale h de l'immeuble saisi au clavier.

Exercice 5 :

Écrire un programme qui lit un nombre au clavier, répond 1 si le nombre est impair et 0 si le nombre est pair.

Exercice 6 :

Écrire un programme qui affiche le chiffre des dizaines d'un nombre saisi au clavier. Même question pour les centaines.

Exercice 7 :

Écrire un programme qui calcule la longueur L d'un câble entre deux pylônes, grâce à la formule :

$$L = a\left(1 + \frac{2}{3}\left(\frac{2f}{a}\right)^2\right)$$

On a « a » est la distance entre les pylônes et « f » la flèche mesuré perpendiculairement au milieu du câble. Ces deux paramètres seront donnés par l'utilisateur.

Exercice 8 :

Écrire un programme qui permet de lire un nombre entier de secondes au clavier, et l'exprimer sous forme d'heures, minutes et seconde en utilisant trois variables : S, M, H.

CORRECTION D'EXERCICES :

Exercice 1 :

1.

```
#include <stdio.h>

int main()
{
	float F,C;
	printf("Veuillez saisir la temperature en Fahrenheit : ");
	scanf("%f",&F);
	C=0.55556*(F-32);
	printf("La temperature en Celsius est %f",C);
	return 0;
}
```

2.

```
#include <stdio.h>

int main()
{
	float F,C;
	printf("Veuillez saisir la temperature en Celsius : ");
	scanf("%f",&C);
	F=C/0.55556+32;
	printf("La temperature en Fahrenheit est %f",F);
	return 0;
}
```

Exercice 2 :

```
#include <stdio.h>

int main()
{
	float PRIX;
	printf("Veuillez saisir le prix d'un composant : ");
	scanf("%f",&PRIX);
	PRIX=PRIX*(1-0.1);
	printf("Le prix du composant en reduction est %f",PRIX);
	return 0;
}
```

Exercice 3 :

1.

```
#include <stdio.h>

int main()
{
	float x,fx;
	printf("Veuillez saisir un nombre : ");
	scanf("%f",&x);
	fx=(2*x+3)*(3*x*x+2);
	printf("L'image de la fonction f est %f",fx);
	return 0;
}
```

2.

```
#include <stdio.h>
```

```
int main()
{
	float x,h,fx,fxh,f_prime;
	printf("Veuillez saisir un nombre : ");
	scanf("%f",&x);
	printf("Veuillez saisir le h : ");
	scanf("%f",&h);
	fx=(2*x+3)*(3*x*x+2);
	fxh=(2*(x+h)+3)*(3*(x+h)*(x+h)+2);
	f_prime=(double)(fxh-fx)/h;
	printf("La derivee de la fonction f est %f",f_prime);
	return 0;
}
```

Exercice 4 :

1.

```
#include <stdio.h>
#define g 9.81

int main()
{
	float h , t ;
	printf("Veuillez saisir la duree de la chute en seconde : ");
	scanf("%f",&t);
	h=0.5*g*t*t;
	printf("La hauteur descendue au bout d'un temps %fs est %fm",t,h);
	return 0;
}
```

2.

```
#include <stdio.h>
#include <math.h>
#define g 9.81

int main()
{
	float h , t ;
	printf("Veuillez saisir la hauteur descendue en metre : ");
	scanf("%f",&h);
	t=(double)sqrt((2*h)/2);
	printf("La duree totale de la chute est %f",t);
	return 0;
}
```

Exercice 5 :

```
#include <stdio.h>

int main()
{
	int n;
	printf("Veuillez saisir un nombre entier : ");
	scanf("%d",&n);
	n=n%2;
	printf("La parite du nombre saisie est %d",n);
	return 0;
}
```

Exercice 6 :

```
#include <stdio.h>

int main()
{
	int n,d,c;
	printf("Veuillez saisir un nombre : ");
	scanf("%d",&n);
	d=(n/10)%10;
	c=(n/100)%10;
	printf("Le chiffre des dizaines est %d et des centaines est %d",d,c);
	return 0;
}
```

Exercice 7 :

```
#include <stdio.h>
#include <math.h>

int main()
{
	int a,f;
	float L;
	printf("Entrez a la distance entre les pylones : ");
	scanf("%d",&a);
	printf("Entrez f la fleche mesure : ");
	scanf("%d",&f);
	L=a*(1+(2.0/3.0)*pow((2.0*f)/a,2)); //pow(x,n) : x puissance n//
	printf("La largeur du cable est %f",L);
	return 0;
}
```

Exercice 8 :

```
#include <stdio.h>

int main()
{
	int S,M,H;
	printf("Entrez un nombre en seconde : ");
	scanf("%d",&S);
	H=S/3600;
	S=S%3600;
	M=S/60;
	S=S%60;
	printf("%d h %d min %d s",H,M,S);
	return 0;
}
```

Chapitre 2 : Les conditions

ÉNONCÉS D'EXERCICES :

Exercice 1 :

Écrire un programme qui lit trois variables au clavier et affiche le maximum des trois.

Exercice 2 :

Écrire un programme permettant de résoudre l'équation **$ax^2+bx+c=0$** en fonction des valeurs de a, b et c.

Exercice 3 :

Écrire un programme qui lit deux nombres entiers a et b et donne le choix à l'utilisateur :

1. de savoir si la somme a + b est paire ;
2. de savoir si le produit ab est pair ;
3. de connaître le signe de la somme a + b ;
4. de connaître le signe du produit ab.

Exercice 4 :

Écrire un programme qui lit la moyenne d'un étudiant et affiche la mention correspondante. La mention est définie comme suit :

Moyenne>=16	T.Bien
14<=Moyenne<16	Bien
12<=Moyenne<14	A.Bien
10<=Moyenne<12	Passable

Exercice 5 :

Écrire un programme qui affiche, selon l'heure saisie, un des messages suivants :

- o Bonjour apparaît lorsque l'heure est comprise entre 0 et 18 heures.
- o Bonsoir de 18h à 22h.
- o Bonne nuit après 22h.

Modifier le programme de telle sorte qu'il prenne en compte le choix de la langue avec laquelle il va afficher le message. Exemple : l'utilisateur choisit 'f' pour afficher en français et 'a' en anglais.

- o Good Morning apparaît lorsque l'heure est comprise entre 0 et 18 heures.
- o Good Evening de 18h à 22h.
- o Good Night après 22h.

Exercice 6 :

Écrivez un programme qui lit deux valeurs entières (A et B) au clavier et qui affiche le signe du produit de A et B sans faire la multiplication.

Exercice 7 :

Écrire un programme qui permet de saisir un entier de trois chiffres non nuls, de déterminer et d'afficher tous les nombres qui peuvent êtres formés par les chiffres de N.

Exemple : N=427

Les nombres formés par les chiffres de N sont : 427, 472, 724, 742, 247, 274.

Exercice 8 :

Écrire un programme qui saisit le numéro du mois et affiche le nombre de jours de ce mois. Pour le cas de février, on lit l'année, si l'année est bissextile, le nombre de jours est 29, sinon c'est 28 jours.

L'Année saisie doit être compris entre 1900 et 2100.

Exercice 9 :

Écrire un programme qui lit deux variables au clavier et les affiche dans l'ordre croissant, quitte à les modifier.

CORRECTION D'EXERCICES :

Exercice 1 :

```
#include <stdio.h>

int main()
{
        float x,y,z,max ;
        printf("Veuillez saisir 3 nombres : ");
        scanf("%f %f %f",&x,&y,&z);
        if (x>y)
        max=x;
        else
        max=y;
        if (z>max)
        max=z;
        printf("Le maximum de ces 3 nombres est %f",max);
        return 0;
}
```

Exercice 2 :

```
#include <stdio.h>
#include <math.h>

int main()
{
        float a,b,c,x,x1,x2,delta;
        printf("Veuillez saisir les coefficients a,b et c de l'equation ax^2+bx+c=0 : ");
        scanf("%f %f %f",&a,&b,&c);
        if (a==0)
        {
          if(b==0)
          {
           if(c==0)
           {
           printf("Tout reel est une solution");
           }
           else
           {
           printf("L'equation n'admet aucune solution");
           }
          }
          else
          {
          x=(double)-c/b;
          printf("La solution de l'equation est %f",x);
          }
        }
        else
        {
        delta=b*b-4*a*c;
          if (delta<0)
          {
          printf("L'equation n'admet aucune solution");
          }
```

```
        else
        {
          if (delta==0)
          {
          x=(double)-b/(2*a);
          printf("La solution de l'equation est %f",x);
          }
          else
          {
          x1=(double)(-b-sqrt(delta))/(2*a);
          x2=(double)(-b+sqrt(delta))/(2*a);
          printf("L'equation admet deux solutions sont %f et %f",x1,x2);
          }
        }
    }
    return 0;
}
```

Exercice 3 :

```
#include <stdio.h>

int main()
{
    int a,b,choix;
    printf("Veuillez saisir le premier nombre entier SVP : ");
    scanf("%d",&a);
    printf("Veuillez saisir le deuxieme nombre entier SVP : ");
    scanf("%d",&b);
    printf("Pour savoir si la somme est paire, appuiyez sur 1 \n");
    printf("Pour savoir si le produit est paire, appuiyez sur 2 \n");
    printf("Pour savoir si le signe de la somme, appuiyez sur 3 \n");
    printf("Pour savoir si le signe du produit, appuiyez sur 4 \n");
    printf("Votre choix SVP : ");
    scanf("%d",&choix);
    switch (choix)
    {
      case 1:
            if ((a+b)%2==0)
            printf("La somme est paire");
            else
           printf("La somme est impair");
           break;
      case 2:
            if ((a*b)%2==0)
            printf("Le produit est paire");
            else
            printf("Le produit est impair");
           break;
      case 3:
            if((a+b)>=0)
            printf("La somme est positive");
            else
            printf("La somme est negative");
            break;
      case 4:
            if((a*b)>=0)
```

```
                printf("Le produit est positif");
                else
                printf("Le produit est negatif");
                break;
        default:
                printf("Erreur...");
                break;
        }
        return 0;
}
```

Exercice 4 :

```
#include <stdio.h>

int main()
{
        float M;
        printf("Entrez votre moyenne SVP : ");
        scanf("%f",&M);
        if(M>=16)
        {
                printf("T.Bien");
        }
        else if(14<=M && M>14)
        {
                printf("Bien");
        }
        else if(12<=M && M<14)
        {
                printf("A.Bien");
        }
        else if(10<=M && M<12)
        {
                printf("Passable");
        }
        else
        {
                printf("Non valide :(");
        }
        return 0;
}
```

Exercice 5 :

```
#include <stdio.h>

int main()
{
        int h;
        char l;
        printf("=====Choix de langue=====\n");
        printf("Appuyez sur 'f' pour afficher le message en francais\n");
        printf("Appuyez sur 'a' pour afficher le message en anglais\n");
        scanf("%c",&l);
        switch (l)
        {
        case 'f':
                printf("entrer svp l'heure entre 0 et 24 SVP : ");
```

```
        scanf("%d",&h);
                if(0<h && h<=18)
                {
                        printf("Bonjour");
                }
                else if(18<h && h<=22)
                {
                        printf("Bonsoir");
                }
                else if(22<h && h<=24 || h==0)
                {
                        printf("Bonne nuit");
                }
                else
                {
                        printf("Erreur...!");
                }
                break;
        case 'a' :
                printf("please enter the hour between 0 and 24 please : ");
        scanf("%d",&h);
                if(0<h && h<=18)
                {
                        printf("Good morning");
                }
                else if(18<h && h<=22)
                {
                        printf("Good evening");
                }
                else if(22<h && h<=24 || h==0)
                {
                        printf("Good night");
                }
                else
                {
                        printf("Error...!");
                }
                break;
        default :
                printf("Appuyez sur 'a' ou 'f' SVP");
          break;
        }
        return 0;
}
```

Exercice 6 :

```
#include <stdio.h>

int main()
{
        int A,B;
        printf("Entrez deux valeurs entieres SVP : ");
        scanf("%d %d",&A,&B);
        if(A==0 || B==0)
        {
                printf("Le produit de %d et %d est nul",A,B);
```

```
	}
	else if((A>0 && B>0) || (A<0 && B<0))
	{
		printf("Le produit est positif");
	}
	else if((A<0 && B>0) || (A>0 && B<0))
	{
		printf("Le produit est negatif");
	}
	return 0;
}
```

Exercice 7 :

```
#include <stdio.h>

int main()
{
	int n,u,d,c,n2,n3,n4,n5,n6;
	printf("Entrez un entier de trois chiffres non nuls SVP : ");
	scanf("%d",&n);
	if(n>=100 && n<=999)
	{
		c=n/100;
		d=(n%100)/10;
		u=n%10;
		if(u!=0 && d!=0 && c!=0)
		{
			n2=c*100+u*10+d;
			n3=d*100+c*10+c;
			n4=d*100+u*10+c;
			n5=u*100+c*10+d;
			n6=u*100+d*10+c;
			printf("%d,%d,%d,%d,%d,%d",n,n2,n3,n4,n5,n6);
		}
		else
		{
			printf("Le nombre ne doit pas comporter un zero");
		}
	}
	else
	{
		printf("Le nombre doit etre compris entre 100 et 999");
	}
	return 0;
}
```

Exercice 8 :

```
#include<stdio.h>

int main()
{
   int annee;
   printf("Saisir une date entre 1900 et 2100 SVP : ");
   scanf("%d",&annee);
   if(annee >= 1900 || annee <= 2100)
   {
      if((annee % 4 == 0 && annee % 100 != 0) || annee % 400 == 0)
```

```
            printf("%d est une annee bissextile\n",annee);
        else
        {
            printf("%d n'est pas une annee bissextile\n",annee);
        }
    }
    return 0;
}
```

Exercice 9 :

```
#include <stdio.h>

int main()
{
        float x,y,temp ;
        printf("Veuillez saisir le premier nombre : ");
        scanf("%f",&x);
        printf("Veuillez saisir le deuxieme nombre : ");
        scanf("%f",&y);
        if (x<y)
        {
                temp=y;
                y=x;
                x=temp;
        }
        printf("%f<%f",y,x);
        return 0;
}
```

Chapitre 3 : Les boucles

ÉNONCÉS D'EXERCICES :

Exercice 1 :

Ecrire un programme faisant calculer et afficher le **factoriel** d'un entier naturel N donné. Sachant que (pour N>0) : N! = N x (N-1) x (N-2) x ...... 3 x 2 x 1.

Exercice 2 :

Ecrire un programme qui permet de lire un entier N et calcule la **somme** des entiers **impairs** inférieurs à N.

Exercice 3 :

1. Ecrire un programme qui demande successivement 20 nombres à l'utilisateur, et affiche le **maximum** de ces 20 nombres :

2. Modifier le programme pour que le programme affiche en quelle **position** avait été saisie ce nombre.

3. Modifier le programme afin qu'il affiche le maximum d'une suite saisie au clavier qui se termine par 0.

Exercice 4 :

Ecrire un programme qui lit deux entiers positifs non nuls a et b et qui calcule le **PGCD** de ces deux nombres en utilisant l'algorithme d'Euclide sachant que :

PGCD (a, b) = PGCD (b, r) avec r =a mod b

Exemple PGCD (32, 12) = PGCD (12, 8)= PGCD (8, 4)=4

Exercice 5 :

La suite de Fibonacci est définie par :

$$\begin{cases} F_0 = 1 \\ F_1 = 1 \\ F_i = F_{i-1} + F_{i-2} \ \ pour\ i \geq 2 \end{cases}$$

Ecrire un programme qui lit un entier positif n et calcule le **n^{ème} terme** de la suite.

Exercice 6 :

Ecrire un programme qui affiche toutes les possibilités d'obtenir un total de **15** en ajoutant **trois entiers** choisis entre 1 et 9 .

Exercice 7 :

Ecrire l'algorithme permettant d'afficher le triangle suivant, le nombre de lignes étant donné par l'utilisateur :

```
1
12
123
1234
12345
123456
```

Exercice 8 :

Ecrire un programme permettant d'afficher le triangle d'étoiles suivant :

Exemple : (pour nL=5)

```
    *
   ***
  *****
 *******
*********
```

Exercice 9 :

Ecrire la un programme qui permet d'afficher un ensemble d'étoiles sous cette forme :

```
*
**
***
****
*****
****
***
**
*
```

Exercice 10 :

Ecrire un programme qui permet de déterminer la **somme des chiffres** d'un nombre entier donné (Exemple : pour N=25418, on aura 2+5+4+1+8=20)

Exercice 11 :

Un entier naturel de trois chiffres est dit **cubique** s'il est égal à la somme des cubes de ses trois chiffres.
Exemple : 153 est cubique car $153=1^3+5^3+3^3$.
Ecrire un programme qui cherche et affiche tous les entiers cubiques de trois chiffres.

Exercice 12 :

Pour un entier naturel n donné, écrire un programme qui calcule la suite :

$$S = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!}$$

Exercice 13 :

Ecrire un programme qui calcule et affiche la valeur de π au moyen de la série suivante (100 termes) :

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots$$

Exercice 14 :

On donne un entier naturel n strictement positif et on définit la suite de Syracuse par :

$$\begin{cases} S_0 = n \\ S_k = S_{k-1} div 2 & \textbf{Si } S_{k-1} \textbf{ est pair} \\ S_k = 3S_{k-1} + 1 & \textbf{Si } S_{k-1} \textbf{ est impair} \end{cases}$$

Ecrire un programme qui lit un entier positif n et calcule le $n^{ème}$ terme de la suite.

Exercice 15 :

Ecrire un programme qui permet de calculer la somme : $S = x + \frac{x^3}{3} + \frac{x^5}{5} + ... + \frac{x^{2n+1}}{2n+1}$.

Exercice 16 :

Ecrire un programme permettant d'afficher le pyramide de nombres suivant :

Exemple d'exécution :

```
Veuillez saisir le nombre des lignes : 6

          1
         232
        34543
       4567654
      567898765
     67890109876
```

Exercice 17 :

Ecrire un programme qui donne une approximation de :

$$e^x \approx 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

Exercice 18 :

Ecrire un programme permettant d'afficher le triangle d'étoiles suivant :

Exemple : (pour nL=5)

```
*********
 *******
  *****
   ***
    *
```

Exercice 19 :

Écrire un programme qui calcule la racine carré d'un nombre réel positif A en utilisant la suite de Newton :

$$\begin{cases} X_0 = A \\ X_{n+1} = \frac{1}{2}\left(X_n + \frac{A}{X_n}\right) \end{cases}$$

Le calcul s'arrête quand $|X_{n+1} - X_n| < \varepsilon$ avec ε donné.

Nous posons $X = X_n$ et $Y = X_{n+1}$.

CORRECTION D'EXERCICES :

Exercice 1 :

```
#include <stdio.h>

int main()
{
	int n,i,fact ;
	printf("Veuillez saisir un nombre entier SVP : ");
	scanf("%d",&n);
	i=1;
	fact=1;
	while (i<=n)
	{
		fact*=i;
		i++;
	}
	printf("Le factoriel de %d est %d",n,fact);
	return 0;
}
```

Exercice 2 :

```
#include <stdio.h>

int main()
{
	int N,S,i;
	printf("Veuillez saisir un nombre entier SVP : ");
	scanf("%d",&N);
	S=0;
	i=1;
	while (i<=N)
	{
		S=S+i;
		i=i+2;
	}
	printf("La somme des entiers impairs inferieurs a %d est %d",N,S);
	return 0;
}
```

Exercice 3 :

1.

```
#include <stdio.h>

int main()
{
	int n,i,max ;
	i=1;
	max=0;
	while (i<=20)
	{
		printf("Veuillez saisir le nombre numero %d : ",i);
		scanf("%d",&n);
		if (n>=max)
		{
			max=n;
		}
		i++;
```

```
	}
	printf("Le maximum de ces nombres est %d",max);
	return 0;
}
```
2.

```
#include <stdio.h>

int main()
{
	int n,i,max,imax ;
	i=1;
	max=0;
	while (i<=20)
	{
		printf("Veuillez saisir le nombre numero %d : ",i);
		scanf("%d",&n);
		if (n>=max)
		{
			max=n;
			imax=i;
		}
		i++;
	}
	printf("Le maximum de ces nombres est %d et sa position est %d" ,max,imax);
	return 0;
}
```
3.

```
#include <stdio.h>

int main()
{
	int n,i,max,imax ;
	i=1;
	max=0;
	while (i<=20)
	{
		printf("Veuillez saisir le nombre numero %d : ",i);
		scanf("%d",&n);
		if (n>=max && (n%10)==0)
		{
			max=n;
			imax=i;
		}
		i++;
	}
	printf("Le maximum des nombres qui se termine par 0 est %d et sa position est %d" ,max,imax);
	return 0;
}
```
Exercice 4 :

```
#include <stdio.h>

int main()
{
	int a,b,r;
	printf("Veuillez saisir deux nombres entiers positifs non nuls SVP : ");
```

```
	scanf("%d %d",&a,&b);
	r=a%b;
	while(r!=0)
	{
		a=b;
		b=r;
		r=a%b;
	}
	printf("Le PGCD de ces deux nombres est %d",b);
	return 0;
}
```

Exercice 5 :

```
#include <stdio.h>

int main()
{
	int u,v,w,i,n;
	printf("Veuillez saisir un entier positif : ");
	scanf("%d",&n);
	u=1;
	v=1;
	i=2;
	while (i<=(n))
	{
		w=u+v;
		v=u;
		u=w;
		i++;
	}
	printf("Le %deme terme de la suite est %d",n,w);
	return 0;
}
```

Exercice 6 :

```
#include <stdio.h>

int main()
{
	int i,j,k;
	for(i=1;i<=9;i++)
	{
		for(j=1;j<=9;j++)
		{
			for(k=1;k<=9;k++)
			if(i+j+k==15)
			{
				printf("%d,%d,%d\n",i,j,k);
			}
		}
	}
	return 0;
}
```

Exercice 7 :

```
#include <stdio.h>

int main()
```

```
{
	int i,j,n;
	printf("Veuillez saisir le nombre des lignes SVP : ");
	scanf("%d",&n);
	for(i=1;i<=n;i++)
	{
		for(j=1;j<=i;j++)
		{
			printf("%d",j);
		}
		printf("\n");
	}
	return 0;
}
```

Exercice 8 :

```
#include <stdio.h>

int main()
{
	int n,i,j,k;
	printf("Veuillez saisir le nombre des lignes : ");
	scanf("%d",&n);
	for (i=1;i<=n;i++)
	{
		for (j=1;j<=n-i;j++)
		{
		   printf(" ");
		}
		for (k=1;k<=2*i-1;k++)
		{
			printf("*");
		}
		printf("\n");
	}
	return 0;
}
```

Exercice 9 :

```
#include <stdio.h>

int main()
{
	int i,j,n;
	printf("Veuillez saisir le nombre des lignes SVP : ");
	scanf("%d",&n);
	for(i=1;i<=n;i++)
	{
		for(j=1;j<=i;j++)
		{
			printf("*");
		}
		printf("\n");
	}
	for(i=n-1;i>=1;i--)
	{
		for(j=1;j<=i;j++)
```

```
			{
				printf("*");
			}
			printf("\n");
		}
		return 0;
}
```

Exercice 10 :

```
#include <stdio.h>

int main()
{
		int n,r,S;
		printf("Veuillez saisir un nombre entier SVP : ");
		scanf("%d",&n);
		S=0;
		while(n!=0)
		{
			r=n%10;
			S=S+r;
			n=n/10;
		}
		printf("La somme des chiffres ce nombre est %d",S);
		return 0;
}
```

Exercice 11 :

```
#include <stdio.h>

int main()
{
		int u,d,c,n;
		for(n=100;n<=999;n++)
		{
			c=n/100;
		d=(n%100)/10;
		   u=n%10;
			if(u*u*u+d*d*d+c*c*c==n)
			{
				printf("%d \n",n);
			}
		}
		return 0;
}
```

Exercice 12 :

```
#include <stdio.h>

int main()
{
		int n,i;
		float S,F;
		printf("Veuillez saisir un entier naturel SVP : ");
		scanf("%d",&n);
		F=1;
		S=1;
		for (i=1;i<=n;i++)
```

```
	{
		F=F*(1.0/i);
		S=S+F;
	}
	printf("La somme de cette suite est %f",S);
	return 0;
}
```

Exercice 13 :

```
#include <stdio.h>

int main()
{
	int i,k,n;
	float pi;
	printf("Veuillez saisir un nombre entier SVP : ");
	scanf("%d",&n);
	pi=0;
	k=1;
	for(i=1;i<=n;i+=2)
	{
		pi=pi+4*k*(1.0/(i));
		k=-k;
	}
	printf("La valeur de pi vaut %f",pi);
	return 0;
}
```

Exercice 14 :

```
#include <stdio.h>

int main()
{
	int n,i,S;
	printf("Veuillez saisir un entier naturel strictement positif : ");
	scanf("%d",&n);
	S=n;
	for(i=1;i<=n;i++)
	{
		if(S%2==0)
		{
			S=S*(1.0/2.0);
		}
		else
		{
			S=3*S+1;
		}
	}
	printf("Le %deme terme de la suite est %d",n,S);
	return 0;
}
```

Exercice 15 :

```
#include <stdio.h>

int main()
{
	int x,n,P,i;
```

```
	float S;
	printf("Veuillez saisir le nombre x : ");
	scanf("%d",&x);
	printf("Veuillez saisir le nombre n : ");
	scanf("%d",&n);
	S=0;
	P=1;
	for(i=1;i<=2*n+1;i++)
	{
		P=P*x;
		if (i%2!=0)
		{
			S=S+P*(1.0/i);
		}
	}
	printf("La somme est %f",S);
	return 0;
}
```

Exercice 16 :

```
#include <stdio.h>

int main()
{
	int n,i,j,k,m;
	printf("Veuillez saisir le nombre des lignes SVP : ");
	scanf("%d",&n);
	for(i=0;i<n;i++)
	{
		for(j=1;j<n-i;j++)
		{
			printf(" ");
		}
		for(k=i+1;k<=2*i+1;k++)
		{
			printf("%d",k%10);
		}
		for(k=2*i;k>=i+1;k--)
		{
			printf("%d",k%10);
		}
		printf("\n");
	}
	return 0;
}
```

Exercice 17 :

```
#include <stdio.h>

int main()
{
	int n,i;
	float x,S,T,P,F;
	printf("Veuillez saisir la variable x : ");
	scanf("%f",&x);
	printf("Veuillez saisir l'ordre n : ");
	scanf("%d",&n);
```

```
        S=1;
        T=0;
        P=1;
        F=1;
        for(i=1;i<=n;i++)
        {
                P=P*x;
                F=F*i;
                T=P*((1.0)/F);
                S=S+T;
        }
        printf("La valeur approchee de exp(%f) est %f",x,S);
        return 0;
}
```

Exercice 18 :

```
#include <stdio.h>

int main()
{
        int n,i,j,k;
        printf("Veuillez saisir le nombre des lignes : ");
        scanf("%d",&n);
        for (i=n;i>=1;i--)
        {
                for (j=1;j<=n-i;j++)
                {
                   printf(" ");
                }
                for (k=1;k<=2*i-1;k++)
                {
                        printf("*");
                }
                printf("\n");
        }
        return 0;
}
```

Exercice 19 :

```
#include <stdio.h>
#include <math.h>
int main()
{
        float A,X,Y,eps;
        printf("Entrez un nombre reel positif SVP : ");
        scanf("%f",&A);
        printf("Entrez un epsilon assez petit : ");
        scanf("%f",&eps);
        Y=A;
        do
        {
                X=Y;
                Y=0.5*(X+(1.0*A)/X);
        } while(fabs(Y-X)>=eps); //fabs(x):valeur absolue de x//
        printf("La racine carre de %f est %f",A,Y);
        return 0;
}
```

Chapitre 4 : Les tableaux

ÉNONCÉS D'EXERCICES :

Exercice 1 :

Soit **T** un tableau contenant N entiers (10≤N≤50). On propose d'écrire un programme qui permet de déterminer et d'afficher la valeur **maximale** et la valeur **minimale** de T.

Exercice 2 :

Ecrire un programme permettant la saisie d'un entier N et l'affichage de sa représentation **en binaire.**

Exercice 3 :

Soit T un tableau contenant N entiers (10≤N≤50). On propose d'écrire un programme qui permet d'éclater T en deux tableaux :

TN (contenant les éléments négatifs de T) et TP (contenant les éléments positifs de T)

Exercice 4 :

Le triangle de Pascal est obtenu en considérant que pour une ligne donné :

- Le premier élément et le dernier élément sont égaux à 1.
- Les autres éléments sont obtenus au moyen de la relation : **T[l][c]= T[l-1][c]+ T[l-1][c-1]**

Ecrire un programme permettant de générer et afficher le triangle de Pascal de taille n.

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

Exercice 5 :

On peut représenter un vecteur de l'espace vectoriel R^n à l'aide d'un tableau de n réels. Écrire un programme qui lit deux vecteurs de R^{10} et calcule leur produit scalaire et affiche les deux vecteurs et leur produit scalaire.

Exercice 6 :

Ecrire un programme qui lit les moyennes des étudiants dans un tableau et qui compte le **nombre d'étudiants** ayant une moyenne **supérieure** ou égale à un **seuil** donné.

Exercice 7 :

Ecrire un programme qui teste si une matrice carrée d'ordre n est symétrique et qui retourne « la matrice M est **symétrique** » ou « la matrice M n'est pas symétrique ».

Exercice 8 :

Ecrire un programme qui teste si une matrice est **triangulaire inférieure** est retourne le résultat.

Exercice 9 :

Ecrire un programme qui qui teste si une matrice est **triangulaire supérieure** est retourne le résultat.

Exercice 10 :

Ecrire un programme qui permet de saisir deux matrices carrées réelles, calculer leur somme et afficher le résultat.

Exercice 11 :

Le produit de deux matrices réelles A=(a_{ij}) et B=(b_{ij}) de dimension n est défini par : $c_{ij} = \sum_{k=1}^{k=n} a_{ik} * b_{kj}$.

Ecrire un programme qui permet de saisir deux matrices carrées réelles, calculer leur produit et afficher le résultat.

Exercice 12 :

Ecrire un programme effectuant le décalage des éléments d'un tableau.

Exemple :

-Tableau initial	D	E	C	A	L	A	G	E

-Tableau modifié	E	C	A	L	A	G	E	D

Exercice 13 :

Écrire un programme qui calcule la moyenne des notes de N étudiants dans un examen en classe, avec N saisi au clavier et toutes les notes dans un tableau.

o Le programme doit demander à l'utilisateur le nombre des étudiants (maximum 20 étudiants si plus il faut afficher un message d'erreur).

o Après, en utilisant une boucle **for** l'utilisateur chargera la note de chaque étudiant dans le tableau, en contrôlant à chaque fois si la note est Valide (supérieur à 0 et inférieur à 10).

o Une somme des notes des différents étudiants sera calculé et affiché.

o Enfin, le programme affiche une moyenne des notes des N étudiants dans un examen en classe ainsi que le nombre des notes Valide et Non-Valide.

Exercice 14 :

Une Matrice carrée est une matrice à n lignes et n colonnes. L'opération de transposition consiste à inverser les lignes et les colonnes en effectuant une symétrie par rapport à la diagonale principale de la matrice.

Exemple :

La matrice $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ devient $\begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$.

Ecrire un programme qui permet de saisir une Matrice carrée puis il cherche et affiche son transposée.

Exercice 15 :

Écrire un programme qui reçoit en entrée 20 nombres entiers et les affichera par ordre croissant.

o Avec une boucle for le programme demandera 20 nombres qui seront inséré dans un tableau.

o Après, avec deux boucles **for** chaque élément va être comparé avec tous les autres. Si le nombre pris en considération par le premier cycle **for** est inférieur à celui par la 2éme boucle for, les éléments vont être inversés entre eux en utilisant une variable auxiliaire de support.

o A la fin des deux boucles **for,** le tableau contiendra les éléments ordonnés mode croissant et sera prés pour l'affichage en utilisant une autre boucle **for.**

Exercice 16 :

Ecrire un programme qui permet de rechercher dans une matrice donnée A les éléments qui sont à la fois un maximum sur leur ligne et un minimum sur leur colonne. Ces éléments sont appelés des points-cols. Afficher les positions et les valeurs de tous les points-cols trouvés.

CORRECTION D'EXERCICES :

Exercice 1 :

```
#include <stdio.h>

int main()
{
	int n,i,max,min;
	printf("Entrez la taille du tableau : ");
	scanf("%d",&n);
	int T[n];
	for(i=0;i<n;i++)
	{
		printf("Entrez l'element Numero %d : ",i+1);
		scanf("%d",&T[i]);
	}
	max=T[0];
	min=T[0];
	for(i=1;i<n;i++)
	{
		if(max<T[i])
		{
			max=T[i];
		}
		if(min>T[i])
		{
		   min=T[i];
		}
	}
	printf("La valeur maximale est %d et la valeur minimale est %d",max,min);
	return 0;
}
```

Exercice 2 :

```
#include <stdio.h>

int main()
{
	int n,i,j;
	int T[50];
	printf("Veuillez saisir un nombre entier SVP : ");
	scanf("%d",&n);
	i=0;
	while(n!=0)
	{
		T[i]=n%2;
		n=n/2;
		i++;
	}
	for(j=i-1;j>=0;j--)
	{
		printf("%d",T[j]);
	}
	return 0;
}
```

Exercice 3 :

```
#include <stdio.h>

int main()
{
	int i,j,k,n;
	int TP[100];
	int TN[100];
	do
	{
		printf("Entrez la taille du tableau SVP : ");
	   scanf("%d",&n);
	}
	while(n<10 || n>50);
	int T[n];
	for(i=0;i<n;i++)
	{
		printf("Entrez l'element numero %d : ",i+1);
		scanf("%d",&T[i]);
	}
	i=0;
	j=0;
	k=0;
	while(i<n)
	{
		if(T[i]>0)
		{
			TP[j]=T[i];
			j++;
		}
		else
		{
			TN[k]=T[i];
			k++;
		}
		i++;
	}
	i=0;
	printf("Les positifs : ");
	while(i<j)
	{
		printf("%d \t",TP[i]);
		i++;
	}
	printf("\n");
	i=0;
	printf("Les negatifs : ");
	while(i<k)
	{
		printf("%d \t",TN[i]);
		i++;
	}
	return 0;
}
```

Exercice 4 :

```
#include <stdio.h>

int main()
{
	int n,l,c;
	printf("Entrez la taille du triangle : ");
	scanf("%d",&n);
	int T[n][n];
	T[0][0]=1;
	for(l=1;l<n;l++)
	{
		T[l][0]=1;
	   T[l][l]=1;
	   for(c=1;c<l;c++)
	   {
		T[l][c]=T[l-1][c]+T[l-1][c-1];
		}
	}
	for(l=0;l<n;l++)
	{
		for(c=0;c<=l;c++)
		{
			printf("%2d",T[l][c]); // %2d : espace d'ordre 2 //
		}
		printf("\n");
	}
	return 0;
}
```

Exercice 5 :

```
#include <stdio.h>

int main()
{
	int i;
	int PS;
	int V1[10];
	int V2[10];
	for(i=0;i<10;i++)
	{
		printf("Entrez l'element %d du vecteur 1 : ",i+1);
		scanf("%d",&V1[i]);
	}
	for(i=0;i<10;i++)
	{
		printf("Entrez l'element %d du vecteur 2 : ",i+1);
		scanf("%d",&V2[i]);
	}
	PS=0;
	for(i=0;i<10;i++)
	{
		PS=PS+V1[i]*V2[i];
	}
	for(i=0;i<10;i++)
	{
```

```
		printf("%2d",V1[i]);
	}
	printf("\n");
	for(i=0;i<10;i++)
	{
		printf("%2d",V2[i]);
	}
	printf("\n");
	printf("Le produit scalaire est %d",PS);
	return 0;
}
```

Exercice 6 :

```
#include <stdio.h>

int main()
{
	int n,i,compteur;
	float seuil;
	printf("Veuillez saisir le nombre des etudiants SVP : ");
	scanf("%d",&n);
	float T[n];
	for(i=0;i<n;i++)
	{
		printf("La note de l'etudiant numero %d est : ",i+1);
		scanf("%f",&T[i]);
	}
	printf("Veuillez saisir le seuil : ");
	scanf("%f",&seuil);
	compteur=0;
	for(i=0;i<n;i++)
	{
		if(T[i]>=seuil)
		{
			compteur=compteur+1;
		}
	}
	printf("le nombre des etudiants ayant une moyenne superieure ou egale a %f est %d",seuil,compteur);
	return 0;
}
```

Exercice 7 :

```
#include <stdio.h>

int main()
{
	int i,j,n;
	int sym=1;
	printf("Entrez la dimension de la matrice SVP : ");
	scanf("%d",&n);
	int M[n][n];
	for(i=0;i<n;i++)
	{
		for(j=0;j<n;j++)
		{
			printf("Entrez l'element M[%d][%d] de la matrice : ",i,j);
			scanf("%d",&M[i][j]);
```

```
                }
        }
        for(i=0;i<n;i++)
        {
                for(j=0;j<n;j++)
                {
                        if(M[i][j]!=M[j][i])
                        {
                                sym=0;
                        }
                }
        }
        if(sym)
                {
                        printf("La matrice M est symetrique");
                }
        else
                {
                        printf("La matrice M n'est pas symetrique");
                }
        return 0;
}
```

Exercice 8 :

```
#include <stdio.h>

int main()
{
        int i,j,n;
        int inf=1;
        printf("Entrez la dimension de la matrice SVP : ");
        scanf("%d",&n);
        int M[n][n];
        for(i=0;i<n;i++)
        {
                for(j=0;j<n;j++)
                {
                        printf("Entrez l'element M[%d][%d] de la matrice M : ",i,j);
                        scanf("%d",&M[i][j]);
                }
        }
        for(i=0;i<n-1;i++)
        {
                for(j=i+1;j<n;j++)
                {
                        if(M[i][j]!=0)
                        {
                                inf=0;
                        }
                }
        }
        if(inf)
        {
                printf("La matrice M est inferieure");
        }
        else
```

```
	{
		printf("La matrice M n'est pas inferieure");
	}
	return 0;
}
```

Exercice 9 :

```
#include <stdio.h>

int main()
{
	int i,j,n;
	int sup=1;
	printf("Entrez la dimension de la matrice SVP : ");
	scanf("%d",&n);
	int M[n][n];
	for(i=0;i<n;i++)
	{
		for(j=0;j<n;j++)
		{
			printf("Entrez l'element M[%d][%d] de la matrice M : ",i,j);
			scanf("%d",&M[i][j]);
		}
	}
	for(i=1;i<n;i++)
	{
		for(j=0;j<i;j++)
		{
			if(M[i][j]!=0)
			{
				sup=0;
			}
		}
	}
	if(sup)
	{
		printf("La matrice M est superieure");
	}
	else
	{
		printf("La matrice M n'est pas superieure");
	}
	return 0;
}
```

Exercice 10 :

```
#include <stdio.h>

int main()
{
	int i,j,n;
	printf("Entrez la dimension de la matrice SVP : ");
	scanf("%d",&n);
	int M1[n][n];
	int M2[n][n];
	int S[n][n];
	for(i=0;i<n;i++)
```

```
	{
		for(j=0;j<n;j++)
		{
			printf("Entrez l'element M1[%d][%d] de la matrice M1 : ",i,j);
			scanf("%d",&M1[i][j]);
		}
	}
	for(i=0;i<n;i++)
	{
		for(j=0;j<n;j++)
		{
			printf("Entrez l'element M2[%d][%d] de la matrice M2 : ",i,j);
			scanf("%d",&M2[i][j]);
		}

	}
	for(i=0;i<n;i++)
	{
		for(j=0;j<n;j++)
		{
			S[i][j]=M1[i][j]+M2[i][j];
		}
	}
	for(i=0;i<n;i++)
	{
		for(j=0;j<n;j++)
		{
			printf("%d \t",S[i][j]);
		}
	   printf("\n");
	}
	return 0;
}
```

Exercice 11 :

```
#include <stdio.h>

int main()
{
	int i,j,k,n;
	printf("Entrez la dimension de la matrice SVP : ");
	scanf("%d",&n);
	int M1[n][n];
	int M2[n][n];
	int P[n][n];
	for(i=0;i<n;i++)
	{
		for(j=0;j<n;j++)
		{
			printf("Entrez l'element M1[%d][%d] de la matrice M1 : ",i,j);
			scanf("%d",&M1[i][j]);
		}
	}
	for(i=0;i<n;i++)
	{
		for(j=0;j<n;j++)
```

```
                {
                        printf("Entrez l'element M2[%d][%d] de la matrice M2 : ",i,j);
                        scanf("%d",&M2[i][j]);
                }

        }
        for(i=0;i<n;i++)
        {
                for(j=0;j<n;j++)
                {
                        P[i][j]=0;
                        for(k=0;k<n;k++)
                        {
                                P[i][j]=P[i][j]+M1[i][k]*M2[k][j];
                        }
                }
        }
        for(i=0;i<n;i++)
        {
                for(j=0;j<n;j++)
                {
                        printf("%d \t",P[i][j]);
                }
          printf("\n");
        }
        return 0;
}
```

Exercice 12 :

```
#include <stdio.h>
int main()
{
        int i,n;
        char temp;
        printf("Entrez la taille du tableau SVP : ");
        scanf("%d",&n);
        char T[n];
        for(i=0;i<n;i++)
        {
                printf("Entrez l'element %d du tableau : ",i+1);
                T[i]=getchar(); //Demande à l'utilisateur de saisir une caractère//
                scanf("%c",&T[i]);
        }
        temp=T[0];
        for(i=0;i<n-1;i++)
        {
                T[i]=T[i+1];
        }
        T[n-1]=temp;
        for(i=0;i<n;i++)
        {
                printf("%c \t",T[i]);
        }
        return 0;
}
```

Exercice 13 :

```
#include <stdio.h>

int main()
{
	int i,n,compt1,compt2;
	float S,M;
	printf("Entrez le nombre d'etudiants SVP : ");
	scanf("%d",&n);
	float T[n];
	if(n>20)
	{
		printf("Erreur...");
	}
	else
	{
		compt1=0;
		compt2=0;
		for(i=0;i<n;i++)
		{
			printf("Entrez la note de l'etudiant numero %d : ",i+1);
			scanf("%f",&T[i]);
			if(0<T[i] && T[i]<10)
			{
				printf(" Non valide :( \n");
				compt1++;
			}
			else
			{
				printf(" Valide :) \n");
				compt2++;
			}
		}
		S=0;
		for(i=0;i<n;i++)
		{
			S=S+T[i];
		}
		M=(double)S/n;
		printf("La somme des notes est %f \n",S);
		printf("La moyenne des notes est %f \n",M);
		printf("Le nombre des notes NV est %d \n",compt1);
		printf("Le nombre des notes V est %d \n",compt2);
	}
	return 0;
}
```

Exercice 14 :

```
#include <stdio.h>

int main()
{
	int i,j,n,temp;
	printf("Entrez la dimension de la matrice SVP : ");
	scanf("%d",&n);
	int M[n][n];
```

```
	for(i=0;i<n;i++)
	{
		for(j=0;j<n;j++)
		{
			printf("Entrez l'element M[%d][%d] de la matrice : ",i,j);
		   scanf("%d",&M[i][j]);
		}
	}
	for(i=0;i<n;i++)
	{
		for(j=0;j<n;j++)
		{
			temp=M[i][j];
			M[i][j]=M[j][i];
			M[j][i]=temp;
		}
	}
	for(i=0;i<n;i++)
	{
		for(j=0;j<n;j++)
		{
			printf("%2d",M[j][i]);
		}
		printf("\n");
	}
	return 0;
}
```

Exercice 15 :

```
#include <stdio.h>

int main()
{
	int i,j,k,temp;
	int T[20];
	for(i=0;i<20;i++)
	{
		printf("Entrez l'element numero %d du tableau : ",i+1);
		scanf("%d",&T[i]);
	}
	for(j=0;j<20;j++)
	{
		for(k=0;k<20;k++)
		{
			if(T[k]>T[j])
			{
			   temp=T[k];
			   T[k]=T[j];
			   T[j]=temp;
			}
		}
	}
	for(i=0;i<20;i++)
	{
		printf("%d \n",T[i]);
	}
```

```
            return 0;
}
```

Exercice 16 :

```
#include <stdio.h>

int main()
{
 /* Déclarations */
 int A[50][50];  /* matrice donnée */
 int MAX[50][50];/* matrice indiquant les maxima des lignes   */
 int MIN[50][50];/* matrice indiquant les minima des colonnes */
 int N, M;    /* dimensions des matrices */
 int I, J;    /* indices courants */
 int AIDE;    /* pour la détection des extréma */
 int C;       /* compteur des points-cols */

 /* Saisie des données */
 printf("Nombre de lignes   (max.50) : ");
 scanf("%d", &N );
 printf("Nombre de colonnes (max.50) : ");
 scanf("%d", &M );
 for (I=0; I<N; I++)
    for (J=0; J<M; J++)
        {
         printf("Element[%d][%d] : ",I,J);
         scanf("%d", &A[I][J]);
        }
 /* Affichage de la matrice */
 printf("Matrice donnee :\n");
 for (I=0; I<N; I++)
    {
     for (J=0; J<M; J++)
          printf("%7d", A[I][J]);
     printf("\n");
    }

 /* Construction de la matrice d'aide MAX */
 /* qui indique les positions de tous les */
 /* maxima sur une ligne. */
 for (I=0; I<N; I++)
     {
      /* Recherche du maximum sur la ligne I */
      AIDE=A[I][0];
      for (J=1; J<M; J++)
           if (A[I][J]>AIDE) AIDE=A[I][J];
      /* Marquage de tous les maxima sur la ligne */
      for (J=0; J<M; J++)
           if (A[I][J]==AIDE)  /* ou bien :                  */
               MAX[I][J]=1;   /* MAX[I][J] = (A[I][J]==AIDE) */
           else
               MAX[I][J]=0;
     }
 /* Construction de la matrice d'aide MIN */
 /* qui indique les positions de tous les */
 /* minima sur une colonne. */
```

```
for (J=0; J<M; J++)
   {
    /* Recherche du minimum sur la colonne J */
    AIDE=A[0][J];
    for (I=1; I<N; I++)
        if (A[I][J]<AIDE) AIDE=A[I][J];
    /* Marquage de tous les minima sur la colonne J */
    for (I=0; I<N; I++)
        if (A[I][J]==AIDE)  /* ou bien :                  */
            MIN[I][J]=1;   /* MIN[I][J] = (A[I][J]==AIDE) */
        else
            MIN[I][J]=0;
   }

/* Edition du résultat */
/* Les composantes qui sont marquées comme extréma */
/* dans MAX et dans MIN sont des points-cols. */
printf("Points - cols :\n");
for (C=0, I=0; I<N; I++)
    for (J=0; J<M; J++)
        if (MAX[I][J]&&MIN[I][J])
          {
           C++;
           printf("L'element %d\test un maximum "
                  "sur la ligne %d\n"
                  "            \t et un minimum "
                  "sur la colonne %d\n", A[I][J], I, J);
          }
if (C==0)
   printf("Le tableau ne contient pas de points-cols.\n");
return 0;
}
```

Chapitre 5 : Les fonctions

ÉNONCÉS D'EXERCICES :

Exercice 1 :

Soit **T** un tableau c